(2010). In *Proceedings of the ACM International Symposium on High Performance Distributed Computing (HPDC).* Chicago, IL.

Simultaneous Performance Exploration and Optimized Search with Volunteer Computing

L. Richard Moore Jr.
Lockheed Martin
6030 South Kent Street
Mesa, Arizona 85212-6061
larry.moore@mesa.afmc.af.mil

Matthew Kopala
L-3 Communications
6030 South Kent Street
Mesa, Arizona 85212-6061
matthew.kopala@L-3com.com

Thomas Mielke
Boeing
6030 South Kent Street
Mesa, Arizona 85212-6061
thomas.mielke@mesa.afmc.af.mil

Michael Krusmark
L-3 Communications
6030 South Kent Street
Mesa, Arizona 85212-6061
michael.krusmark@mesa.afmc.af.mil

Kevin A. Gluck
Air Force Research Laboratory
6030 South Kent Street
Mesa, Arizona 85212-6061
kevin.gluck@mesa.afmc.af.mil

ABSTRACT
Volunteer computing is a powerful platform for solving complex scientific problems. MindModeling@Home is a volunteer computing project available to the cognitive modeling community for conducting research to better understand the human mind. We are interested in optimizing search processes on volunteer resources, yet we are also interested in exploring and understanding changes in model performance across interacting, non-linear mechanisms and parameter spaces. To support both of these goals, we have developed a stochastic optimization approach and integrated it with MindModeling@Home. We tested this approach with a cognitive model on a sample parameter space, demonstrating significant decreases in computational resource utilization and search runtime, while also providing useful visual representations of performance surfaces. Future work will focus on scaling the technique to more volunteers and larger parameter spaces, as well as optimizing the performance of the search algorithm in regards to the challenges inherent with volunteer computing.

Categories and Subject Descriptors
G.1.6 [**Mathematics of Computing**]: Optimization – *global optimization, stochastic optimization.*

General Terms
Algorithms and Performance.

Keywords
Stochastic optimization, global optimization, parameter space, volunteer computing, search, BOINC.

1. INTRODUCTION
The cognitive science community aims to understand the nature of the human mind. Towards this end, it is common to develop software-based cognitive models that instantiate a theoretical account of some aspect of human cognition. Exercising the model reveals performance and behavior characteristics that are used for validation, prediction, and prescription.

Many cognitive scientists construct models within the framework of a *cognitive architecture*, which provides theoretically motivated constraints based on years of psychological research. These constraints represent the invariant characteristics of the human cognitive system that are assumed to be relatively persistent and consistent across individuals, contexts, and time. Along with these constraints come architectural parameters that can influence model behavior in meaningful ways, such as influencing the rate at which the model "thinks" or how easily it can recall knowledge [2]. The parameters constitute a search space, and while the number of dimensions and increments across them can vary, most of our spaces are between 100 thousand and 2 million parameter combinations. The time it takes to test a model at a particular parameter combination can vary greatly depending on the task and context, ranging from a fraction of a second to hours. Furthermore, the results are often highly stochastic as a reflection of human performance, and the model may need to be run hundreds of times to determine the central tendency.

Large parameter spaces with long running models require significant amounts of computational resources to search [6]. At the time of this writing, volunteer computing resources contributing to Berkeley Open Infrastructure for Network Computing (BOINC) projects boast 3.1 petaflops of computational power [3], which makes volunteer computing a

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED
JUN 2010	**Conference Proceedings**	**01-01-2009 to 30-04-2009**

4. TITLE AND SUBTITLE	5a. CONTRACT NUMBER
Simultaneous performance exploration and optimized search with volunteer computing	**FA8650-05-D-6502 TO 44**
	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER **61105F**
6. AUTHOR(S) **Richard Moore Jr; Matthew Kopala; Thomas Mielke; Michael Krusmark; Kevin Gluck**	5d. PROJECT NUMBER **2313**
	5e. TASK NUMBER **AC**
	5f. WORK UNIT NUMBER **01**
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **Air Force Research Laboratory/711HPW/RHA,Warfighter Readiness Research Division,6030 South Kent Street,Mesa,AZ,85212-6061**	8. PERFORMING ORGANIZATION REPORT NUMBER **AFRL; 711HPW/RHA**
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) **Air Force Research Laboratory/711HPW/RHA, Warfighter Readiness Research Division, 6030 South Kent Street, Mesa, AZ, 85212-6061**	10. SPONSOR/MONITOR'S ACRONYM(S) **AFRL; AFRL/RHA; 711HPW**
	11. SPONSOR/MONITOR'S REPORT NUMBER(S) **AFRL-RH-AZ-PR-2010-0003**

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES
Published in Proceedings of the ACM International Symposium on High Performance Distributing Computing, held 20-25 Jun 10, in Chicago IL

14. ABSTRACT

Volunteer computing is a powerful platform for solving complex scientific problems. MindModeling@Home is a volunteer computing project available to the cognitive modeling community for conducting research to better understand the human mind. We are interested in optimizing search processes on volunteer resources, yet we are also interested in exploring and understanding changes in model performance across interacting, non-linear mechanisms and parameter spaces. To support both of these goals, we have developed a stochastic optimization approach and integrated it with MindModeling@Home. We tested this approach with a cognitive model on a sample parameter space, demonstrating significant decreases in computational resource utilization and search runtime, while also providing useful visual representations of performance surfaces. Future work will focus on scaling the technique to more volunteers and larger parameter spaces, as well as optimizing the performance of the search algorithm in regards to the challenges inherent with volunteer computing.

15. SUBJECT TERMS **Volunteer computing; MindModeling@Home; Cognitive modeling; Human mind; Model performance; Visual representations; Parameter spaces;**					
16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**	**Public Release**	**4**	

tempting resource for the cognitive modeling community's large computational demands. To avail these resources, we have developed MindModeling@Home [8]

2. MINDMODELING@HOME

The MindModeling@Home system can be used to simulate a cognitive model on a large number of volunteer resources, thereby decreasing the time required to explore and enumerate a parameter space compared to limited, costly local resources. MindModeling@Home is an implementation of a BOINC task server on our own hardware, with the addition of a batch management system, a domain specific client application, and a web interface for model submission. Using the web interface, the modeler uploads their model, specifies the parameter space to be searched, selects the version of the cognitive architecture to be used, and then submits the batch. The batch processing system is responsible for dividing the parameter space into work units, which are then submitted to the BOINC task server.

Next, BOINC clients running on MindModeling@Home volunteer computers download the client application and one or more work units to be computed. The client application is responsible for running the model using the selected cognitive architecture and collecting the results. Once the computation of a work unit is complete, the BOINC client sends the results back to the server.

Upon receiving results from a volunteer resource, the MindModeling@Home batch system processes and aggregates the data. The batch system tracks how much of the search space has been explored, uses this to determine when the job is complete, and presents the batch progress to the modeler via the web interface.

Even with large numbers of volunteer resources, searching cognitive model parameter spaces can still be computationally overwhelming. To reduce the computation demand, we have experimented with intelligent search and exploration algorithms integrated with the MindModeling@Home framework.

3. OPTIMIZATION STRATEGIES

There are many optimization strategies to choose from that work well within the context of a workstation. A smaller set generalizes well to high performance computing (HPC) clusters, and an even smaller set generalizes well to volunteer computing networks.

Volunteer resources present a challenging computational context because volunteers have a great deal of systemic control—they pull down work when they like, and they provide results if and when they like. Yet optimization algorithms by nature are designed to be in control—they measure samples, make a decision, measure more samples, etc. If the optimization algorithm lacks enough completed samples to make a decision—perhaps because a volunteer computer was retasked or shut off--the algorithm cannot move forward, and cannot generate meaningful new work for volunteers until time-outs provoke remedial measures. Parallelization declines, and overall efficiency is lost.

Other BOINC projects have confronted this challenge with various search optimization algorithms. MilkyWay@Home, for example, has developed a parallel genetic algorithm as well as a particle swarm optimization for BOINC [5]. POEM@HOME has published results using several techniques: the stochastic tunneling method, the basin hopping technique, the parallel tempering method, and an evolutionary approach [10].

The optimization techniques used are telling, because they all fall under the broad category of *stochastic optimization.* The stochastic optimization family of techniques intentionally introduce and rely upon random elements in the search process. Because work generated includes random elements, there is no limit to the amount of work available at any time for volunteers (i.e. we can generate limitless random numbers). Furthermore, because decisions are based on stochastic information, these algorithms are typically robust to incomplete data that might result when a volunteer fails to return results in a reasonable amount of time. Stochastic optimization approaches, it seems, are a good match for volunteer resources.

4. CELL

The approaches adopted by other volunteer projects work well for searching, but they do not fully meet our requirements. We are not only interested in searching for optima, but it is also useful in our line of research to visually *explore* the parameter space [7]. This is an important distinction because most optimized search strategies, such as those mentioned above, tend to localize sampling, which makes it difficult to produce a plot of the full parameter space..

Therefore, we have developed a new methodology and implementation called Cell that samples broadly enough in the parameter space to produce visualizations while simultaneously searching for optima. The process is fairly simple: we begin by sampling the entire parameter space with a stochastic uniform distribution. As volunteers return the results of their model runs, Cell estimates the best fitting hyper-plane for each dependent measure via simple linear regression.

A single flat hyper-plane poorly approximates a typical cognitive model parameter space, so once the sample count has reached a critical threshold, the parameter space is split in half along its longest dimension. The critical threshold for splitting is currently defined as 2x the number of samples required to produce good regression predictions, as defined by Knofcyznski and Mundfrom [9],

Following the first split there is a single bend in the space, as each resulting half is independently analyzed for best fitting hyper-planes. At this point the algorithm skews the sampling distribution toward the half of the space that better fits human performance. Once volunteers return enough samples, it too will split, the sampling distribution will again be adjusted, and the process continues until the best fitting section of the space is too small to split (a resolution defined by the modeler). The resulting structure of divisions and analyses is often called a regression tree [1].

While Cell has met with some success on workstation and high performance computing clusters, it was unclear whether the integration challenges with MindModeling@Home would compromise performance (these issues are discussed below). To test Cell's efficiency on MindModeling@Home, we ran the same cognitive model twice, once with Cell and the other as a full combinatorial mesh. The space was comprised of two parameters, each with 51 divisions, producing a mesh of 2601 nodes. Although Cell will sample anywhere, it was configured to split the space along the same grid lines used in the full combinatorial mesh,

Because the cognitive models produce stochastic results, the full combinatorial mesh sampled each node 100 times to obtain a reliable measure of central tendency. While not critical for visual exploration of the full space, an accurate central tendency is necessary for searching for the best model fit. To control the test and measure resource utilization, four dedicated local machines with two cores each substituted for volunteer resources.

5. RESULTS

During our test we tracked CPU utilization, computational efficiency, and wall clock time. We also tested the quality of the search and visualization results. Table 1 shows the difference in performance between the two runs.

Table 1. Performance comparison between the full combinatorial mesh and Cell. Better performance is bolded.

Metric	Full Combinatorial Mesh	Cell
Implementation Efficiency		
Model Runs	260,100	**17,100**
Search Duration (hours)	20.13	**5.23**
Avg. CPU Utilization (Volunteers)	**68.5%**	24.6%
Avg. CPU Utilization (Server)	6.43	**2.59**
Optimization Results		
R – Reaction Time	.97	.97
R – Percent Correct	**.94**	.90
Overall Parameter Space		
RMSE – Reaction Time	**28.9ms**	128.8ms
RMSE – Percent Correct	**.7%**	1.3%

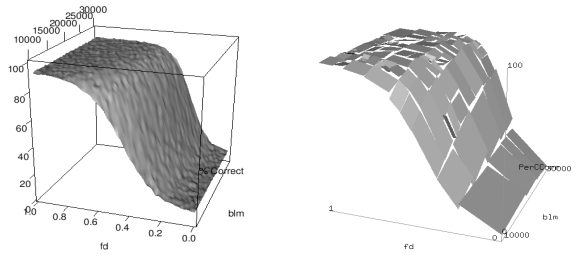

Figure 1. Full combinatorial mesh parameter space, left, compared with the Cell parameter space, right. The best fitting data are towards the top, which is more finely detailed due to more intense sampling.

Cell demonstrated a large computational savings in terms of model runs, requiring only 6.5% of the full combinatorial mesh. Wall clock time was also improved, although not quite as dramatically with a 74% reduction in time.

The volunteers utilized an average of 44% less CPU resources during the course of the Cell run versus the full combinatorial mesh. While this was not unexpected (see Discussion), a higher CPU utilization is desirable to make the most of available resources. On the server side CPU utilization was slightly less with Cell, but additional tests will be required to determine whether the difference is significant and, if so, identify the root cause.

To test the effectiveness of the search algorithm at identifying the optimal fit of the model to human data, we reran the model 100x using the predicted best-fitting parameter values from each approach. We then computed the correlation between model performance and human performance for two key task dependent measures: reaction time and percent correct. Although the full combinatorial mesh produced better results than Cell, Cell's results were still very usable and well worth the computational savings.

For our cognitive modeling work the parameter space outside of the optimal fitting area serves a descriptive purpose, so capturing the qualitative behavior of the parameters is a higher concern than quantitative predictions. Figure 1 shows a comparison of the parameter spaces constructed with full combinatorial mesh versus Cell. We also computed the quantitative difference as shown in Table 1 under the heading "Overall Parameter Space." The RMSD values for the two main dependent measures were calculated by running a second full combinatorial mesh and comparing it to the first full mesh and to interpolated Cell data.

6. DISCUSSION

Although the results are promising, the model only represents one example, and the volunteer count was intentionally limited so that we could monitor resource utilization. We suspect that the optimization benefits will be directly related to the model performance, the number of volunteers, and the size of the work unit (i.e., how many samples are computed) for volunteers.

For example, consider 500 volunteers, all available to search a new model. Traditionally, MindModeling@Home sizes work units to last about an hour, which for a fast model like the one we used could amount to 6000 samples. 500 volunteers with 6000 samples each would require Cell to generate a uniform distribution with 3 million samples to accommodate the available resources. If it only takes 100 samples to make a decision and split the space, there will be approximately (3,000,000 – 100) / 2 samples calculated unnecessarily in the down selected half of the space. The data will still be useful for visualization, or in the event that the search reselects the other half of the space, but generally this is an undesirable behavior.

To mitigate this issue we used small work units for the Cell run, but this approach comes with a price. For fast models like the one used in our test, small work units decrease the computation / communication time ratio on the volunteer resources, thus decreasing efficiency. The smaller ratio with the Cell run is evidenced by the 44% reduction in CPU utilization on the volunteers. Most of our cognitive models are much slower than the one used in this test, however, so in practice the issue may be alleviated or eliminated, depending upon the specific model.

Our approach to integrate Cell with MindModeling@Home required that Cell maintain a stockpile of work for volunteers. It was difficult to keep enough work units ready for distribution, without requesting more samples than were needed to split the space. We set the amount of samples sent out to remain between 4 – 10 times the number required, in consideration that some clients would take longer than others to return results, and to maintain enough work to keep the clients busy computing samples a greater

percentage of the time. This way, although some computational work may have been superfluous, the overall run time decreased, and volunteer requests for new work were fulfilled more frequently. In the future, a tighter integration between Cell and BOINC that generates work dynamically upon request should help address this issue.

RAM utilization is another consideration when using Cell. Because Cell is constantly receiving new data and recomputing regression planes, it must maintain the data in memory for efficiency. In our test, Cell's RAM usage was as expected (about 200 bytes per sample), but even this modest amount can become a limitation with tens of millions of samples.

Although we ran Cell on our servers, the Rosetta@home project, which aims to predict protein structures, uses a different approach. In their case, many volunteers make rough predictions of the same protein structure using a stochastic optimization technique. The results returned include many different predictions with varying degrees of success, and the best prediction is then plucked out from among them [4]. For MindModeling@Home, this approach may be desirable to reduce CPU and memory loads on the servers. In this scenario, Cell would run on the volunteer resources. By reducing the threshold of samples required to split the space, best fits would be predicted much more quickly, albeit more roughly. We could then sift through all the results returned to determine the best overall fit, just like Rosetta@home. This is an interesting option that we may explore in the future.

7. CONCLUSION

Optimizing search and exploration with volunteer computing resources is a challenging problem. Like other volunteer computing projects, we have found that this integration is best achieved through stochastic optimization algorithms. Our algorithm differs from others used in the volunteer computing community in that we characterize the full parameter space as well as searching for best fits. This algorithm was successfully integrated with our volunteer computing project, MindModeling@Home, and the test results were promising.

Future refinement will focus on tuning the relationship between work unit size, model performance, and the amount of volunteer resources available. Understanding this relationship and exploring variations to our current approach will be the focus of future work.

8. ACKNOWLEDGMENTS

The views expressed in this paper are those of the authors and do not reflect the official policy or position of the Department of Defense or the U.S. Government. This research was sponsored by grants 07HE01COR and 10RH04COR from the Air Force Office of Scientific Research, and by the Warfighter Readiness Research Division of the Air Force Research Laboratory's Human Effectiveness Directorate.

9. REFERENCES

[1] Alexander, W. P., & Grimshaw, S. D. 1996. Treed Regression. *Journal of Computational and Graphical Statistics*, 5, 156-175.

[2] Anderson, J. R. 2007. *How can the human mind occur in the physical universe?* Oxford University Press, Oxford, UK.

[3] BOINC, 2010. Retrieved January6, 2010, from Berkeley University of California: http://boinc.berkeley.edu/.

[4] Das R, Qian B, Raman S, Vernon R, Thompson J, Bradley P, Khare S, Tyka MD, Bhat D, Chivian D, Kim DE, Sheffler WH, Malmström L, Wollacott AM, Wang C, Andre I, Baker D. 2007. Structure prediction for CASP7 targets using extensive all-atom refinement with Rosetta@home. *Proteins* 69 Suppl 8, 118-28.

[5] Desell, T., Magdon-Ismail, M., Szymanski, B., Varela, C., Newberg, H. and Cole N. 2009. Robust Asynchronous Optimization for Volunteer Computing Grids. In *5th IEEE International Conference on e-Science* (December 2009) Oxford, UK, 263-270.

[6] Gluck, K. A., Scheutz, M., Gunzelmann, G., Harris, J., and Kershner, J. 2007. Combinatorics meets processing power: Large-scale computational resources for BRIMS. In *Proceedings of the Sixteenth Conference on Behavior Representation in Modeling and Simulation* (Orlando, Florida). Simulation Interoperability Standards Organization, 73-83.

[7] Gluck, K. A., Stanley, C. T., Moore, L. R., Reitter, D., Halbrügge, M. 2010. Exploration for Understanding in Model Comparisons. Under review, *Journal of Artificial General Intelligence.*

[8] Harris, J., Gluck, K. A., Mielke, T., and Moore, L. R. 2009. MindModeling@Home … and Anywhere Else You Have Idle Processors [Abstract]. In A. Howes, D. Peebles, & R. Cooper (Eds.) *Proceedings of the Ninth International Conference on Cognitive Modeling* (Manchester, United Kingdom), paper 249.

[9] Knofcyznski, G. T., & Mundfrom, D. 2008. Sample sizes when using multiple linear regression for prediction. *Educational and Psychological Measurement*. 68, 431-442.

[10] Schug, A., Verma, A., Wenzel, W., and Schoen, G. 2005. Biomolecular structure prediction with stochastic optimization methods. *Adv. Eng. Materials* 7, 1005.